Verification of the Formal Approach to Data Fusion

Ildar R. Baimuratov, Stephan D. Morozov, and Natalia A. Zhukova

ITMO University, Saint Petersburg, Russia
baimuratov.i@gmail.com

Abstract. We present a comprehensive overview of numerous existing data fusion models to show that in substance they are informal. Informal models are hardly applicable in designing real data fusion systems. Therefore, we represent and continue to develop the formal approach to data fusion proposed in our previous works. It includes, first, a formal language based on first order predicate logic and, second, category theoretic model derived from the formal language. We verify our formal approach by formalizing specification of a data fusion system on the logical language and modeling it by means of the category theoretic model.

Keywords: Data fusion · Data fusion models · Data fusion systems · Logic · Category theory.

1 Introduction

Today the fields of data analysis and machine learning are flourishing. But this field is rather a collection of different unlinked methods and techniques than a comprehensive and consistent system like more mature scientific fields. The situation is analogous to one before industrial revolution: there were different handicrafts but inventing mass production integrated separated craftsmen into extensive manufacturing organization and allowed to dramatically increase productive efficiency. Therefore, it is crucial to develop an entire theory for data analysis and machine learning field. In this article we continue an attempt to develop such formal approach.

We considered data fusion, being the process of integrating data, as a source for the required theory and data fusion models as prototypes. We surveyed different data fusion models but it turned out that, unfortunately, none of them is formal enough and grounded on any fundamental mathematical theory. Therefore, we had to develop a formal theory for data fusion, that would be rooted in fundamental mathematics. We consider logic and category theory as such mathematical foundation. The overview of existing data fusion models is presented in the Section 2.

The first step at developing any formal theory is specifying the language that will be used for formalizing subject field. Though contemporary data analysis techniques use different mathematical notions, they hardly could be considered as an integrated formal context cause of its usage inconsistency. Thus, our first goal was to specify a formal language that would be able to comprise all existing techniques. For this purpose, we use the language of first order predicate logic. Specification of the first order predicate logic is presented in the Section 3.1. After that, we analyze the structure of logical constructions be means of category theory in order to derive a formal mathematical model, which intended to be the formal model of data fusion. The resulting model is presented in the Section 3.2.

The purpose of the current work is to examine the proposed formal approach by applying it to description of a real data fusion system. Its application means formalizing the description on predicate logic language. If the description is successfully formalized, it implies that the category theoretic model is verified as the model is derived from the logical language. As result, we achieve a formal model of the considered data fusion system.

2 Existing data fusion models overview

Developing an integrated formal theory of data analysis and machine learning, we considered data fusion models as prototypes. We present an overview of most notable data fusion models.

2.1 JDL fusion model

The JDL [1] was one of the first and most widespread data fusion systems, was created in 1986 by the Joint directors of Laboratories data fusion working group, as a result of a working at the US Department of Defense, to aid the developments in military applications. The model consists of 4 levels [2]

1 **Object refinement**: data integration for getting improved representation of individual objects.
2 **Situation refinement**: description of relations between objects and environment.
3 **Threat refinement**: projecting of current situation in future and predicting of consequences.
4 **Process refinement**: meta-process that monitors data- fusion processes to maintain real-time performance.

2.2 Dasarathy model

Dasarathy model [3] is a Data-Based Model. It is functionality oriented model, rather than the task it is based on fusion function. There are three levels of abstraction:

a) Data
b) Feature
c) Decision

And following function:

- **DAI-DAO (data in — data out)** is the most basic data fusion method. The primary input and output is a raw data. The output is more accurate and reliable. Data fusion is done immediately after the Data ' s gathered from the sensors.
- **DAI-FEO (data in — feature out)**. Extract the features and characteristics of raw data. Feature extraction.
- **FEI-FEO (feature in — feature out)**. Set of features will be worked to improve a feature or extract a new feature.
- **FEI-DEO (feature in — decision out)**. Decision can be taken based on the set of feature as an input. The decision can be Pattern recognition and pattern processing.
- **DEI-DEO (decision in — decision out)**. Obtain a new decisions from the fused decisions. This is a high level fusion

2.3 Boyd control model

Boyd control loop is the activity based model. This is the four stage cyclic loop. This is also called OODA loop, which refers Observe, Orient, Decide, and Act. Developed by the Military Strategist and U.S States Air Force Colonel John Boyd in the year 1987 [4]. There is four stages:

- **Observe**. Collect and pre-process all the data from sensor.
- **Orient**. Collect data is fused to clarify the current scenario.
- **Decide**. Based on the previous stage, (i.e. Orient) action plans are decided for the future.
- **Act**. Where is the plan which is decided in the previous stage is executed.

2.4 Waterfall model

The waterfall model is an hierarchical architecture which divided in 3 levels, each one with two modules. The information outputted by one module will be inputted to the next module:

- Data is gathered from the environment and properly transformed, delivering not only the data processed but also information about the sensors to the next level.
- The main features are extracted from the data from the previous module and fused, thus reducing the quantity of data transmitted and increasing their information richness.
- According to the processing from the previous levels, a scenario of events is created and possible routes of action are assembled. The last module (Decision Making) delivers enough information to the control module to calibrate and configure the sensors.

2.5 Luo and Kay Model

Luo and Kay presented a hierarchical model [7], yet different from the waterfall model. While in the waterfall model all data gathered is processed in a sequential way for all modules, in the Luo and Kay model data from the sensors are incrementally added on different fusion centers (multi sensor fusion), thus increasing the level of representation, from the raw data or signal level to more abstract symbolic representations of the data at the symbol level.

2.6 Intelligence cycle model

The intelligence cycle is also called Intelligence process by U.S Department of Defense and the uniformed services. The intelligence cycle consist from four stage:

- **Collection**. Collect the raw data from the environment.
- **Collation**. All the collected data is analyzed and correlated. Redundant and unreliable data's are discarded.
- **Evaluation**. The collated data' s are fused and analyzed.
- **Dissemination**. Produce the decision based on the result of fused data from the previous stage Evaluation

2.7 Omnibus model

Omnibus model [5] is a hybrid model which presents some features from other model. It comprises a flow chart, a dual-perspective definition and a structured repository of accumulated expertise, and consist from following levels:

- **Observe**. This level measures the environment, gathering and processing data from the sensors, delivering the information to the second level
- **Orientate**. Here, the data is fused and the main features extracted in order to reduce the amount of data.
- **Decide**. The third level concerns with the presentation of the processed data to the human operator or/and act on the environment.
- **Act**. The model is in a closed loop with a control module to calibrate the sensors.

2.8 Thomopoulos Model

Thomopoulos [6] also proposed a three level model, formed by the signal, evidence and dynamics levels. In each level, data gathered is confronted with data previously processed and stored, preserving a given order.

1 **The signal level** process that information, performing correlations due to the in existence of a mathematical model. Therefore, the data gathered is correlated with information previously stored in the database, in a learning process.

2 On **the evidence level**, data is combined at different levels of inference based on a statistical model and the assessment required by the user (e.g. decision making or hypothesis testing).
3 On **the dynamics level** a mathematical model is used to perform the data fusion.

As we see, none of these data fusion models has explicit formal definitions of its elements and none of them has link to any fundamental mathematical theory. Therefore, first of all we had to develop a systematic formal approach to data fusion.

3 Formal approach to data fusion

In [8] we proposed the formal framework for data fusion. It consists of the logic language and the category theoretic model.

3.1 Logic language

The logic language for formalizing data fusion is in fact classic first order predicate logic. It consists of

- individual variables: $x_1, x_2, ...$;
- individual functions: $f_1, f_2, ...$;
- predicate functions: $P_1, P_2, ...$;
- propositional variables: $A_1, A_2, ...$;
- propositional functions: $\neg, \wedge, \vee, \rightarrow, ...$;
- quantifiers: $\forall, \exists$.

Each individual function is n-placed, if $n = 0$, f is an individual constant. Similarly, if propositional function is 0-placed, it is the propositional constant $\bot$ or $\top$. Quantifier $\forall$ is considered as generalized conjunction and $\exists$ is considered as generalized disjunction.

The notion of object in predicate logic corresponds to the notion of term. Terms are strings of symbols that are constructed according to the following inductive definition:

- a variable x is a term,
- if f is an n-placed function and $t_1, ..., t_n$ are terms, then $f(t_1, ..., t_m)$ is a term.

Facts are denoted by formulas. There are atomic and complex formulas:

- if ϕ is an n-placed predicate P and $t_1, ..., t_n$ are terms then $P(t_1, ..., t_n)$ is an atomic formula;
- if x is a variable and ϕ, ψ are formulas then
  - $\forall x \phi$ and $\exists x \phi$;
  - $\neg \phi$;

- $\phi \wedge \psi, \phi \vee \psi, \phi \rightarrow \psi$;

are complex formulas.

The language of predicate logic is interpreted by models. A model $\mathcal{M}$ consists of a universe A, a nonempty set of objects, and an interpretation function $\mathcal{I}$, which assigns structures of the universe A to structures of the language:

– $\mathcal{I}(x) = X$, where $X \subseteq A$;
– $\mathcal{I}(f^n) = A^n \rightarrow A$;
– $\mathcal{I}(P^n) = A^n \rightarrow \{0, 1\}$.

The fact that a formula ϕ corresponds to truth is denoted by the notation $\mathcal{M} \models \phi$:

$$\mathcal{I}(\phi) = 1 \Leftrightarrow \mathcal{M} \models \phi.$$

Therefore, if ϕ is an atomic sentence of the form $P(t_1, ..., t_n)$,

$$\mathcal{M} \models P(t_1, ..., t_n) \Leftrightarrow \mathcal{I}(P(t_1, ..., t_n)) = 1.$$

If ϕ is a complex sentence:

– $\mathcal{M} \models \forall x \phi \Leftrightarrow$ for every $a \in X$ it is true that $\mathcal{I}(\phi) = 1$;
– $\mathcal{M} \models \exists x \phi \Leftrightarrow$ exists $a \in X$, such that $\mathcal{I}(\phi) = 1$;
– $\mathcal{M} \models \neg\phi \Leftrightarrow \mathcal{M} \not\models \phi$;
– $\mathcal{M} \models \phi \wedge \psi \Leftrightarrow \mathcal{M} \models \phi$ and $\mathcal{M} \models \psi$;
– $\mathcal{M} \models \phi \vee \psi \Leftrightarrow \mathcal{M} \models \phi$ or $\mathcal{M} \models \psi$;
– $\mathcal{M} \models \phi \rightarrow \psi \Leftrightarrow \mathcal{M} \not\models \phi$ or $\mathcal{M} \models \psi$.

otherwise $\mathcal{I}(\phi) = 0$.

3.2 Category theoretic model

Finally, we obtain the formal model for data fusion generalizing logical structures by means of category theory. Analyzing definitions listed above we extract two classes of objects:

– the universe A;
– truth-values $TV = \{0, 1\}$;

and three classes of mappings:

– terms $Term$: $A \rightarrow A$;
– predicates $Pred$: $A \rightarrow TV$;
– propositions $Prop$: $TV \rightarrow TV$;

where variables are considered as identity mappings

$$id_X : X \rightarrow X$$

and given mappings $f : X \to Y$ and $g : Y \to Z$ there is a composition

$$(f \circ g) : X \to Z$$

for any objects X, Y, Z and mappings f, g. Resulting model is represented by the following diagram:

$$\begin{tikzcd} \mathrm{A} \arrow[loop left, "\mathsf{Term}"] \arrow[r, "\mathsf{Pred}"] & \mathrm{TV} \arrow[loop right, "\mathsf{Prop}"] \end{tikzcd}$$

We are going to verify our formal model expressing every logical structure in category theoretical terms:

- individual variables: $id_A : A \to A$;
- individual functions: $A \to A$;
- predicate functions: $A \to TV$;
- propositional variables: $id_{TV} : TV \to TV$;
- propositional functions: $TV \to TV$;
- quantifiers: $TV \to TV$.

As complex logical structures are defined inductively, any further construction is expressible in category theoretical terms as well.

4 Data fusion system formalization

In order to verify our formal approach, we are going to consider an example of data fusion system and apply the approach to its specification.

4.1 Specification of the data fusion system

Consider the following data fusion system [9]: Two data structures are given that include three attributes (id (number), value(text/number), value(text/number)). Data comes from a variety of sources, and for their comparison, a simple rule for allowing entities is used up. Also a simple mapping of entities by coincidence of values is used. In general, the experimental model consists of:

1. **Deduplicated collection** is an input collection, like $A(id, a, b, c)$ or $B(id, a, b, d)$
2. **Minimum Union operator** is defined as the combination of outer union with the subsequent removal of subsumed tuple.
3. **Outer Union operation** results in the union of two relations. If the schemes do not match, the resulting scheme is the union of the two original. Outer Union, in fact, is implemented using the FusionIndex index.
4. **FusionIndex**. For each id, all relevant records are obtained and those that are absorbed by them are deleted
5. **subsumption**. The tuple $t1$ absorbs another tuple $t2$, if they:
   - have the same schemes;
   - in $t2$ there are more unknown (null) values than in $t1$;
   - in $t2$ all known values coincide with the values in $t1$.

6. **is_subsumed** function. Checks whether one tuple is absorbed by a pairwise comparison of attributes or a null test.
7. **removeSubsumed** function. Deletes all the captured records from the tuple.
8. **minUnion**. It is necessary for construction of resulting tuples at realization of Minimum Union.
9. **Data Fusion** operator. The Data Fusion operator is a special kind of function that uses grouping to overcome conflicts. The basic idea is to group different representations of the same entity by a common attribute, and then to apply the conflict resolution functions to all other attributes. There are two types of strategy for conflict resolution functions:
   - The deciding strategy is to select some one value in some way (minimum, maximum, random value);
   - The mediating strategy is the aggregation of all values (mean, sum).
10. Fused collection is the resulting collection.

For example, let us take two collections: $A(id, a, b, c)$ and $B(id, a, b, d)$. Attributes a, b, c, d can contain null values, Attribute id is the same. Below is an example of similar data for the collection
$A(id, name, age)$

$$\{id : 7600, name : null, age : null\}, \{id : 1500, name : zvk, age : 938\}$$

$B(id, name, info)$

$$\{id : 1500, name : null, info : null\}, \{id : 7600, name : pjg, info : null\}$$

Also there are functions of calculating the mean, choosing a random nonzero value, and concatenating. This merging of input collections according to specified rules forms the Fused collection, with an average value for the age attribute, any non-zero value for the name and for the attribute info, all the available values will be concatenated. For the concerned collections the resulting collection is
$C(id, name, age, info)$

{id:1500, name:zvk, age:938, info:null}, {id:7600, name:pjg, age:null, info:null}.

4.2 Logical formalization

Analyzing structure of the data fusion system, we extract following elements:

- Symbols: id, a, b, c;
- Input collections: $A(id, a, b)$, $B(id, a, c)$;
- Operations: $Minimumunion$ and $Datafusion$;
- Resulting collection: $C(id, a, b, c)$.

To formalize these elements we will define to what kind of logical structures they belong.

- Symbols, being the simplest elements of the system, are individual constants, or 0-placed individual functions. Therefore, symbol variables are individual variables, or identity mappings for individuals.
- Collections are structures that have individuals as arguments. As collections do not map individuals to other individuals, but only integrate them into complex structures, they are predicates. For all values of argument id, being a primary key, predicate corresponding to a collection should be defined, therefore, the argument id is binded with the quantifier $\forall$. All other parameters are binded with the quantifier $\exists$.
- Operation $Minimumunion(MU)$ joins two collections into one, i.e. it maps collections to collections, therefore, $Minimumunion$ is a propositional function.
- Operation $Datafusion$ is a variable for some conflict resolution function for individuals. These functions are $min, max, random, mean, sum$. All these functions map individuals to individuals, therefore, $Datafusion$ is a variable for individual functions. And as variables are identity mappings $Datafusion$ is individual function as well.

In result, the described data fusion system can be represented as the following formula:

$$\forall id((\exists a \exists b A(id, a, b) \wedge \exists a \exists c B(id, a, c)) \rightarrow \\ \exists a \exists b \exists c C(id, mean(a), random(b), conc(c))).$$

4.3 Category theoretic modeling

We are going to compare the result of formalization of the described data analysis system with the proposed category theoretical data fusion model. In result of formalization we have the following logical structures:

- individual variables id, a, b, c;
- individual functions $mean, random, conc$;
- predicates $A(id, a, b), B(id, a, c), C(id, a, b, c)$;
- quantifiers $\forall id, \exists a, \exists b, \exists c$;
- propositional function $\rightarrow_{MU}$.

According to the category theoretic classification of logical structures, there are following mappings:

- terms:
  - individual variables: id, a, b, c;
  - individual functions: $mean, random, conc$;
- predicates: $A(id, a, b), B(id, a, c), C(id, a, b, c)$;
- propositions:
  - quantifiers: $\forall id, \exists a, \exists b, \exists c$;
  - operation $\rightarrow_{MU}$.

Summing up, the result of formalization is represented in the Table 1 and the following diagram:

Table 1. The result of formalization

Category-theory	Logic	System	Example
Term	Individual variable	Symbols	id, a, b, c
Term	Individual function	Data fusion	$mean, random, conc$
Predicate	Predicate	Collection	$A(id, a, b), B(id, a, c), C(id, a, b, c)$
Proposition	Quantifier	—	data structure
Proposition	Propositional function	*Minimum union*	$\rightarrow_{MU}$

$$\begin{tikzcd}
\text{Symbols} \arrow[loop left, "\text{Data fusion}"] \arrow[r, "\text{Collections}"] & \text{TV} \arrow[loop right, "\text{Quantifiers, Minimum union}"]
\end{tikzcd}$$

String variables, being identical mapping, is omitted.

It is notable, that we have quantifiers, which formalize the data structure, not mentioned explicitly in the description of the data fusion system. The fact that our formal system exposed them demonstrates the advantage of our formal approach.

5 Conclusion

Developing the formal approach to data fusion, we surveyed most notable data fusion models and found out that none of them is formal enough. We represented the formal approach proposed in the previous works, which includes formal language based on the first order predicate logic and category theoretic model derived from the formal language. After that, in order to verify the proposed approach we considered an example of data fusion system, formalized it with logical language and derived the category theoretic model of the system from the result of formalization. This fact demonstrates applicability of our formal approach. Moreover, it turned our that comprehensiveness of our formal approach allows to expose elements of the system not mentioned explicitly in its specification.

References

1. Steinberg, A.N., Bowman, C.L., and White, Jr., F.E.: Revisions to the JDL Data Fusion Model. Proc. 3rd NATO/IRIS Conf., Quebec City, (1998)
2. Steinberg, A.N., Bowman, C.L.: Revisions to the JDL data fusion model. In: Handbook of Multisensor Data Fusion, pp. 45–67 (2009)
3. Dasarathy, B.: Decision Fusion. Proc. IEEE Computer Society Press, (1994)
4. Fadok, D.S., Boyd, J., and Warden, J.: Air Power's Quest for Strategic Paralysis, Maxwell Air Force Base AI. Air University Press, (AD-A291621) (1995)
5. Bedworth, M. and O Brien, J.: The Omnibus model: a new model of data fusion. DERA Malvern preprint, 1999
6. Thomopoulos, S. C.: Sensor integration and data fusion. Proc. SPIE 1198, Sensor Fusion II: Human and Machine Strategies, 178–191 (1989)

7. Luo, R. C., Kay, M. G.L.: Multisensor Integration and Fusion for Intelligent Machines and Systems. Ablex Publishing Corp 1995
8. Baimuratov, I.R., Zhukova, N.A.: A formal framework for Data Fusion. International Journal of Applied Mathematics and Informatics **11**, 56–64 (2017)
9. А. Е. Вовченко 1 , Л. А. Калиниченко 2 , Д. Ю. Ковалев: МЕТОДЫ РАЗРЕШЕНИЯ СУЩНОСТЕЙ И СЛИЯНИЯ ДАННЫХ В ETL-ПРОЦЕССЕ И ИХ РЕАЛИЗАЦИЯ В СРЕДЕ HADOOP. Информатика и ее применение Т 8. вып 94—109 (2014)